

University of Groningen

A new identification method for fuzzy linear models of nonlinear dynamic systems

Bruin, H.A.E. de; Roffel, B.

Published in:
Journal of Process Control

DOI:
[10.1016/0959-1524\(96\)00003-0](https://doi.org/10.1016/0959-1524(96)00003-0)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
1996

[Link to publication in University of Groningen/UMCG research database](#)

Citation for published version (APA):

Bruin, H. A. E. D., & Roffel, B. (1996). A new identification method for fuzzy linear models of nonlinear dynamic systems. *Journal of Process Control*, 6(5). [https://doi.org/10.1016/0959-1524\(96\)00003-0](https://doi.org/10.1016/0959-1524(96)00003-0)

Copyright

Other than for strictly personal use, it is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license (like Creative Commons).

The publication may also be distributed here under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license. More information can be found on the University of Groningen website: <https://www.rug.nl/library/open-access/self-archiving-pure/taverne-amendment>.

Take-down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Downloaded from the University of Groningen/UMCG research database (Pure): <http://www.rug.nl/research/portal>. For technical reasons the number of authors shown on this cover page is limited to 10 maximum.

A new identification method for fuzzy linear models of nonlinear dynamic systems

H. A. E. de Bruin* and B. Roffel†

University of Twente, Faculty of Chemical Engineering, PO Box 217, 7500AE Enschede, The Netherlands

Received 17 July 1995; revised 17 January 1996

The most promising methods for identifying a fuzzy model are data clustering, cluster merging and subsequent projection of the clusters on the input variable space. This article proposes to modify this procedure by adding a cluster rotation step, and a method for the direct calculation of the consequence parameters of the fuzzy linear model. These two additional steps make the model identification procedure more accurate and limits the loss of information during the identification procedure. The proposed method has been tested on a nonlinear first order model and a nonlinear model of a bioreactor and results are very promising.

Keywords: fuzzy linear model; model identification; fuzzy clustering

During the past decades considerable progress has been made in our ability to model, identify and control complex systems. Identification and control of linear systems is well established and also certain classes of nonlinear systems have been analyzed in detail. Nonlinear systems, however, have such a variety of possible structures, that no representation is likely to be universally valid for their identification from experimental data. Many approaches to this problem have been pursued, some of which are apparently general but limited in practice due to computational complexity. What may be needed to make further progress is a change in orientation or approach; more specifically the use of soft computing, rather than hard computing.

Soft computing is tolerant of imprecision, uncertainty and partial truth; its principal constituents are fuzzy logic, neurocomputing and probabilistic reasoning.

The nonlinear dynamic system which will be the focus of this investigation can be described by the following discrete nonlinear regression model:

$$y_{k+1} = f(y_k, \dots, y_{k-n+1}, u_k, \dots, u_{k-m-d+1}) \quad (1)$$

where k is the discrete time, m and n are model orders, d is a pure delay, y is the process output and u is the process input. Equation (1) can also be written more conveniently as:

$$y_{k+1} = f(D^k) \quad (2)$$

where D^k is the regression vector

$$D^k = \{u_i, y_j \mid i = k - m - d + 1 \dots k, j = k - n + 1 \dots k\} \quad (3)$$

The function of f in Equation (2) describes a hyper surface; identification of the model means approximation of this surface. Appendix I illustrates this concept with an example. The key idea in fuzzy modeling is to use a simple linguistic description of the process instead of precise mathematical relations between the variables. A fuzzy model can be regarded as a nonlinear mapping between discrete, mutually overlapping regions in model input and output spaces. Several approaches to fuzzy modeling have been presented in the literature. There are three distinct classes of fuzzy models. *Fuzzy relational models*¹⁻⁴ are based on the theory of fuzzy relations and relational equations. *Fuzzy linguistic models*^{5,6} and *fuzzy linear models*⁷⁻⁹ make use of IF-THEN rules.

Fuzzy relational models

Fuzzy relational models use fuzzy relations to capture associations between fuzzy regions in system input, state and output domains. Formally a fuzzy relational model of a single input–single output (SISO) system can be written as a generalization of a standard difference equation

*Present address: AKZO-Nobel Information Services, Velperweg 76, 6824 BM Arnhem, The Netherlands.

†Author to whom correspondence should be addressed.

$$Y(k) = U(k-1) \cdot U(k-2) \cdot \dots \cdot U(k-m) \cdot Y(k-1) \cdot \dots \cdot Y(k-n) \cdot R \quad (4)$$

where $U(k)$ and $Y(k)$ are the fuzzy representation of the input $u(k)$ and the output $y(k)$ at time k with respect to reference fuzzy sets defined in the input-output universe; m and n are model orders. R is a multi-dimensional fuzzy relation

$$R : \underbrace{U \times \dots \times U}_{m \text{ times}} \times \underbrace{Y \times \dots \times Y}_{n \text{ times}} \rightarrow [0,1] \quad (5)$$

defined in Cartesian product of the input and output universes U and Y . The \cdot symbol denotes relational composition. The algorithm to calculate the model output is:

1. Fuzzification, find the degree of membership of inputs, past inputs and past outputs in the reference fuzzy sets.
2. Calculate the output fuzzy set, using relational composition [Equation (4)].
3. Defuzzification of the output fuzzy set, using usually the fuzzy-mean method.

As an example¹⁰, consider a SISO model $Y = X \cdot R$ with three reference fuzzy sets in both input and output (e.g. LOW, MEDIUM, HIGH) and the fuzzy relation:

$$R = \begin{bmatrix} 0.9 & 0.4 & 0.1 \\ 0.3 & 1.0 & 0.1 \\ 0.0 & 0.6 & 1.0 \end{bmatrix} \quad (6)$$

This relation defines three rules, each with three weighted consequents. For instance, the first row of the relation matrix reads:

$$R^1: \text{ if } x \text{ is LOW then } y \text{ is LOW (0.9), MEDIUM (0.4), HIGH (0.1)}$$

Fuzzy linguistic models

The idea of linguistic modeling was introduced in the pioneering papers of Zadeh¹¹ and applied by Mamdani¹² to fuzzy control of dynamical processes. Therefore, this class of fuzzy model is also called Mamdani's model, but in this article the term linguistic model is used consistently. In a typical linguistic rule

$$\text{if } \underbrace{x_1 \text{ is } \textit{SMALL} \text{ and } x_2 \text{ is } \textit{LARGE}}_{\text{premise}} \dots \text{ then } \underbrace{y \text{ is } \textit{BIG}}_{\text{consequent}} \quad (7)$$

both the rule premise and the consequents are defined by means of fuzzy sets. To analyze the models, the theory of fuzzy relations can be applied.

Fuzzy linear models

A class of fuzzy models introduced by Takagi and Sugeno⁷ provides an effective way to represent nonlin-

ear systems, by combining a rule-based description with a local functional description (usually linear regression models). Though various names are used in the literature (Takagi-Sugeno-Kang or TSK model, quasi-linear model, etc.), in this article the model is consistently named fuzzy linear model. Typical rules for a dynamic system may look as follows:

$$R^1: \text{ if } u(k-1) \text{ is } \textit{small} \text{ and } y(k-1) \text{ is } \textit{medium} \text{ then } y^1(k) = p_1^1 + p_2^1 u(k-1) + p_3^1 y(k-1)$$

$$R^2: \text{ if } u(k-1) \text{ is } \textit{large} \text{ and } y(k-1) \text{ is } \textit{large} \text{ then } y^2(k) = p_1^2 + p_2^2 u(k-1) + p_3^2 y(k-1) \quad (8)$$

where $u(k-1)$ and $y(k-1)$ are premise variables, linguistic terms like *SMALL*, *LARGE*, etc, are defined as fuzzy sets, p_j are consequent parameters different for each rule.

A generalized i th rule of a fuzzy linear model of a MISO system will look like:

$$R^i: \text{ if } \mathbf{U}^i = \mathbf{A}^i \text{ and } \mathbf{y}^i = \mathbf{b}^i \text{ then } y(k) = p_0 + \sum_{j=1}^m p_j^u u_1(k-j) + \dots + \sum_{j=1}^m p_{j+(N-1)*m}^u u_n(k-j) + \sum_{l=1}^n p_l^y y(k-l) \quad (9)$$

where $\mathbf{y}^i = [y^i(k-1), \dots, y^i(k-n)]$ is a vector with fuzzified past outputs, $y(k)$ the model output, $u(k-j)$ and $y(k-1)$ the past inputs and output, n and m are the model orders, N is the number of inputs, p^u and p^y are the consequence parameters of the inputs and output, \mathbf{A}^i is a matrix with fuzzy sets, \mathbf{b}^i is a vector with fuzzy sets, and

$$\mathbf{U} = \begin{bmatrix} u_1(k-1) & u_1(k-2) & \dots & u_1(k-m) \\ u_2(k-1) & u_2(k-2) & \dots & u_2(k-m) \\ \vdots & \vdots & \ddots & \vdots \\ u_N(k-1) & u_N(k-2) & \dots & u_N(k-m) \end{bmatrix} \quad (10)$$

is a matrix with all fuzzified past inputs. The overall membership degree of the premise of rule R^i can be calculated as:

$$\mu_i = \min(\mathbf{A}^i, \mathbf{b}^i) \quad (11)$$

The model output is computed as a weighted mean of the individual rule contributions:

$$y = \frac{\sum_{i=1}^K \mu_i y_i}{\sum_{i=1}^K \mu_i} \quad (12)$$

where μ_i is the overall membership degree of the i th rule premise, y_i is the consequent value of the i th rule and K is the total number of rules.

As one may expect, it is more difficult to identify fuzzy relational models and fuzzy linguistic models than fuzzy linear models from numerical data, since the structureless nature — similar to a look-up table — of the former two types of models provide a less effective representation. This means that more rules may be needed to approximate a function to any given degree of accuracy than with a fuzzy linear model.

Model identification

In the following sections a new identification method will be discussed, based on fuzzy clustering.

The main idea underlying the use of fuzzy clustering for the identification of nonlinear systems is very simple. In the product space of model inputs and output, the behaviour of a nonlinear dynamic MISO system forms a hypersurface. The model inputs are the recent discrete-time history of the systems inputs $u_i(k)$, $u_i(k-1)$, $u_i(k-2)$, ..., $u_i(k-m)$ and the past outputs $y(k-1)$, $y(k-2)$, ..., $y(k-n)$, where m and n are the model orders. Fuzzy clustering is used to fit an *a priori* specified number of hyperplanes through this nonlinear surface, approximating the system by a collection of local linear models. The theoretical background of this technique has been described by Yoshinari *et al.*¹³ and Yager and Filev¹⁴; Zhao *et al.*^{9,15} also present some simulations.

Fuzzy clustering

Fuzzy clustering is an important tool to identify the structure in data. In general, a fuzzy clustering algorithm with objective function can be formulated as follows: let $X = \{x^j | j = 1, 2, \dots, N\}$ be a finite set of feature vectors in \mathbb{R}^n , where N is the number of measurements and n is the dimension of the input variables, $x_j = [x_1^j, x_2^j, \dots, x_n^j]^T$ and $P = (P_1, P_2, \dots, P_K)$ be a K -tuple of prototypes each of which characterizes one of the K clusters; a partition of X into K fuzzy clusters will be performed by minimizing the objective function

$$J(P, U; X) = \sum_{i=1}^K \sum_{j=1}^N (u_{ij})^m d^2(x_j, P_i) \quad (13)$$

where $U = [u_{ij}]_{K \times N}$, $u_{ij} \in [0, 1]$, is called a fuzzy K -partition matrix and satisfies the following conditions:

$$0 < \sum_{j=1}^N u_{ij} < N, \forall i \quad \sum_{i=1}^K u_{ij} < 1, \forall j \quad (14)$$

u_{ij} represents the membership grade of feature point x_j for cluster P_i , $d(x_j, P_i)$ is the distance from a feature point x_j to cluster P_i , $m \in [1, \infty]$ is a weighting exponent. In words, this means that the input-output product space X ($N \times n$ matrix) is divided into K clusters. If the fuzzy exponent $m > 1$ then each feature belongs to each cluster with a membership value $u = [0, 1]$. The

total of all membership values of a feature is equal to 1. A cluster can have different shapes, depending on the choice of prototypes. The calculation of the membership values is dependent on the definition of the distance measure. Thus if a feature is closer to a cluster, the membership value of this feature to this cluster will be higher.

According to the choice of prototypes and the definition of the distance measure, different fuzzy clustering algorithms are obtained. If the prototype of a cluster is a point — the cluster center — it will give spherical clusters, if the prototype is a line it will give tubular clusters, and so on. In view of the linear form of the consequence part in linear fuzzy models, an obvious choice of fuzzy clustering is the fuzzy C-varieties (FCV) algorithm, in which linear or planar clusters are allowed as prototypes to be sought. Another algorithm, fuzzy C-elliptotypes (FCE) has a convex combination of distance measures. Both algorithms, however, are not suitable for seeking linear or planar clusters¹⁶, because they seek infinitely spanned linear and planar clusters. The Gustafson–Kessel (GK) algorithm is one of the clustering algorithms which are the most appropriate for linear or planar clusters.

Gustafson–Kessel (GK) clustering algorithm

The GK algorithm¹⁷, which is the fuzzy generalization of the Adaptive Distance Dynamic Clusters algorithm⁹, searches for ellipsoidal clusters. It can be used for linear or planar clusters because this type of cluster can be viewed as a special case of ellipsoids for which one or more radii are zero.

In the GK algorithm, the distance from a point x_j to a cluster P_i is

$$d^2(x_j, P_i) = (x_j - v_i)^T M_i (x_j - v_i) \quad (15)$$

where v_i and $M_i = |F_i|^{1/n} F_i^{-1}$ are the cluster center and a positive-definite symmetric matrix related to the covariance matrix F_i of the i th prototype, and n is the dimension of the input-output product space. The GK algorithm characterizes the geometric structures of clusters better than other algorithms used in identification of fuzzy models^{13,16,18}. The shapes of the i th cluster can be described, to some extent, by the scatter matrix¹⁹

$$S_i = \sum_{j=1}^N (u_{ij})^m (M_i^{1/2} (x_j - v_i))(M_i^{1/2} (x_j - v_i))^T \quad (16)$$

If the set of data concentrated around the center forms an ellipsoidal shaped cluster, then the principal axes of the ellipsoid will be given approximately by the eigenvectors of S_i , and the relative length of the axes by the corresponding eigenvalues.

The GK algorithm may be formulated as follows: given a set of data $\{x_j | j = 1, 2, \dots, N\}$ and initial guesses for cluster center v_i , covariance matrix F_i , and the fuzzy partition matrix $U = [u_{ij}]$, the following steps can be performed iteratively:

1. compute the distances

$$d^2(x_j, P_i) = (x_j - v_i)^T M_i (x_j - v_i) \quad (15)$$

2. compute the membership values

$$u_{ij} = \frac{[d^2(x_j - v_i)]^{-1/m-1}}{\sum_{l=1}^K [d^2(x_j - v_l)]^{-1/m-1}} \quad (17)$$

if $d^2(x_j, P_i) = 0$ for some $i = k$, set $u_{kj} = 1$, and $\forall i \neq k$, $u_{ij} = 0$.

3. compute new cluster centers

$$v_i = \frac{\sum_{j=1}^N u_{ij}^m x_j}{\sum_{j=1}^N u_{ij}^m} \quad (18)$$

4. compute new covariance matrices

$$F_i = \frac{\sum_{j=1}^N u_{ij}^m (x_j - v_i)(x_j - v_i)^T}{\sum_{j=1}^N u_{ij}^m} \quad (19)$$

until a specified convergence criterion is satisfied, e.g. $\max |\Delta u_{ij}| < \varepsilon$. The cluster center matrix is randomly initialized, and the covariance matrix is initialized with the identity matrix. The fuzzy GK clustering uses two parameters, a weighting exponent m ($m = [1, \infty]$, for a crisp model $m = 1$, fuzzy model $m > 1$ but mostly $m = 2$) also called the fuzzy exponent, and the number of clusters K .

Modified compatible cluster merging (MCCM) algorithm

Making a fuzzy model of a dynamic nonlinear process requires the tuning of many parameters. Doing this heuristically is tedious and time consuming. Clustering techniques provide an easier way for forming fuzzy models using measurements made on the system. However, the number of clusters, the number of rules in the rule base, must be determined *a priori*. In order to limit the number of rules, it is recommended to merge clusters which show a certain degree of conformity. For this purpose Kaymak²⁰ developed a new algorithm called the modified compatible cluster merging (MCCM) algorithm. The key elements of this algorithm are criteria which measure the degree of compatibility between clusters. These criteria are based on the cluster distances and eigenvectors of the covariance matrices of the clusters which the GK algorithm finds. Let the centers of two clusters be v_i and v_j . Let the eigenvectors of the two

clusters be $\{\phi_{i,1}, \dots, \phi_{i,n}\}$ and $\{\phi_{j,1}, \dots, \phi_{j,n}\}$. It is assumed that the vectors are arranged in descending order. The following criteria are defined for merging clusters:

$$|\phi_{i,n} \cdot \phi_{j,n}| \geq c1_{ij}, \quad c1_{ij} \text{ close to } 1 \quad (20)$$

$$\|v_i - v_j\| \leq c2_{ij}, \quad c2_{ij} \text{ close to } 0 \quad (21)$$

Since the samples in a cluster lie approximately on a hyperplane, the first criterion states that the clusters should be merged if the hyperplanes are almost parallel. The second criterion states that the clusters should be sufficiently close to be merged. By evaluating these criteria for all pairs of clusters, one obtains the matrices $C1[c1_{ij}]$ and $C2[c2_{ij}]$ whose elements indicate the degree of similarity between the i th and j th clusters measured according to the corresponding criterion. The aforementioned two criteria are by themselves not sufficient for successfully establishing which clusters should be merged. Therefore a decision making algorithm is used. The algorithm takes as its inputs the matrices $C1$ and $C2$ and maps every element onto a two-dimensional space using two membership functions, resulting in the matrices $\tilde{C}1$ and $\tilde{C}2$. The membership functions indicate the degree of compatibility between two clusters, based on the evidence from the corresponding criterion.

The membership functions, which are of the exponential type, are depicted in Figure 1. The points aa and bb which limit the support of the membership functions are calculated as follows:

$$aa = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N C1, \quad bb = \frac{1}{N(N-1)} \sum_{i=1}^N \sum_{\substack{j=1 \\ j \neq i}}^N C2, \quad (22)$$

and the exponential membership functions are:

$$\mu_1 = e^{\left(\frac{-7(c1-1)^2}{(1-aa)^2}\right)} \quad \text{and} \quad \mu_2 = e^{\left(\frac{-7c2^2}{-bb^2}\right)} \quad (23)$$

A coefficient equal to 7 is used in Equation (23) since exponential functions never result in a membership of zero ($e^{-7} \cong 0.001$).

Notice that the criteria of parallelity and closeness of clusters may partially compensate each other. In other words, two clusters that are slightly non-parallel but very close may need to be merged. The same applies to the clusters which are parallel but somewhat far from each other. Taking this fact into account, the elements of the matrices $\tilde{C}1$ and $\tilde{C}2$ are combined to form the matrix $S[\mu_{ij}]$ using the geometric mean as the decision operator:

$$\mu_{ij} = \sqrt{\tilde{C}1_{i,j} \tilde{C}2_{i,j}} \quad (24)$$

This operator allows for partial compensation between the criteria [Equations (20) and (21)]. Thus one obtains the matrix $S[\mu_{ij}]$ whose elements $\mu_{ij} \in [0, 1]$

indicate the overall degree of compatibility between the clusters i and j . The diagonal elements of S are by definition equal to 1.

Merging clusters

The next step is to determine which clusters are to be merged, given the similarity matrix S . This is done by using the fuzzy relational clustering method²¹. In this method, a matrix of similarity measures is converged to a solution by employing the well-known max-min composition several times. The resulting matrix is thresholded with a predetermined and problem dependent threshold value γ ($\gamma = [0, 1]$, 0 means clusters will always be merged, 1 means clusters will never be merged). This results in a partition of the clusters and from this information one may determine the candidate groups of clusters to be merged.

For example, given the matrices $C1$ and $C2$ [see Equations (20) and (21)]:

$$C1 = \begin{bmatrix} 1 & 0.95 & 0.2 & 0.1 \\ 0.95 & 1 & 0.3 & 0.1 \\ 0.2 & 0.3 & 1 & 0.4 \\ 0.1 & 0.1 & 0.4 & 1 \end{bmatrix} \text{ and } C2 = \begin{bmatrix} 0 & 1 & 6 & 2 \\ 1 & 0 & 1 & 7 \\ 6 & 1 & 0 & 9 \\ 2 & 7 & 9 & 0 \end{bmatrix}$$

aa and bb can be calculated: $aa = 0.34$ and $bb = 4.33$. With the membership functions μ_1 and μ_2 the matrices $\tilde{C}1$ and $\tilde{C}2$ can be calculated, this is the fuzzification step. This will result in the following matrices $\tilde{C}1$ and $\tilde{C}2$:

$$\tilde{C}1 = \begin{bmatrix} 1 & 0.95 & 0 & 0 \\ 0.95 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.18 \\ 0 & 0 & 0.18 & 1 \end{bmatrix}$$

$$\text{and } \tilde{C}2 = \begin{bmatrix} 1 & 0.69 & 0 & 0.23 \\ 0.69 & 1 & 0.69 & 0 \\ 0 & 0.69 & 1 & 0 \\ 0.23 & 0 & 0 & 1 \end{bmatrix}$$

By using the geometric mean as operator between the matrices $\tilde{C}1$ and $\tilde{C}2$ the similarity matrix S can be calculated:

$$S = \begin{bmatrix} 1 & 0.81 & 0 & 0 \\ 0.81 & 1 & 0.01 & 0 \\ 0 & 0.01 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

With a threshold value $\gamma = 0.8$ only clusters 1 and 2 are candidates for merging. The modified cluster merging technique is used and merging compatibility conditions are checked with the threshold $\gamma = 0.8$. Kaymak²⁰ recommends to choose the threshold between 0.75 and 0.55, depending on the nonlinearity of the system. To

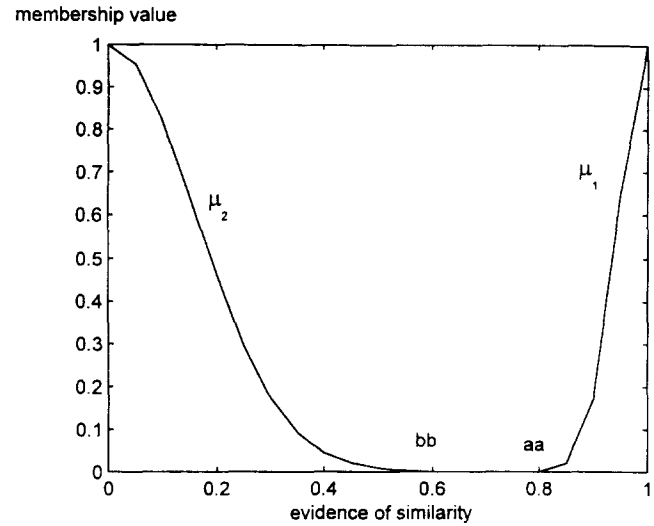


Figure 1 The membership functions μ_1 and μ_2 with $aa = 0.8$ and $bb = 0.6$

check if the threshold of $\gamma = 0.8$ is reasonable, first a test run is done with a threshold of $\gamma = 0.01$ (this means that after every epoch always two clusters will be merged). Before each cluster merging, the similarity matrix is calculated. The two clusters with the highest similarity index will be merged. If one plots all the highest similarity indices versus the number of clusters, one will find a curve like the one shown in Figure 2. It is advisable to choose the threshold to be the same as the similarity index of the clusters at the inflection point of the curve.

Membership rotation and projection

After clustering and cluster merging, one has obtained the optimal number of clusters, hence the number of rules in the rulebase. The validity regions of the clusters must be extracted from the clustering results. These validity regions are encapsulated within the fuzzy K -partition matrix [Equation (13)]. In order to extract the regions, the input-output product space has to be projected onto the input product space. Hence one dimension is lost: the output. Now the membership function must be retrieved from the fuzzy K -partition matrix. To do so, both Zhao *et al.*⁹ and Babuška and Verbruggen¹⁰ project the membership of each feature onto the input vari-

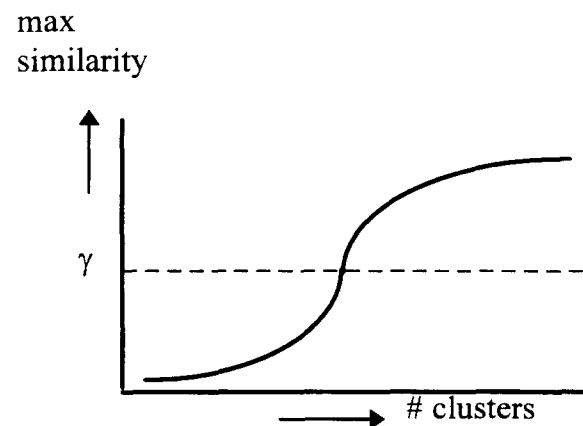


Figure 2 Maximal similarity of two clusters versus the number of clusters

ables. Zhao *et al.* use a least squares method to fit the projection into a trapezoidal membership function. Babuška and Verbruggen, however, use a low pass filter to filter the noise, followed by a least squares method to fit the projection in an exponential membership function. Both methods may give serious problems when a cluster is not orientated in the direction of an input variable.

Projection on the input product space can result in the loss of information. The loss will be minimal if a cluster is orientated in the main direction of the input space. The main direction of a cluster is indicated by the largest eigenvector of the covariance matrix of this cluster. The loss of information is maximal if the cluster is not orientated as one of the input variables. In that case reconstruction of the projected membership functions will represent a cluster which is maximally different from the original cluster.

It is very likely that the fuzzy clusters have an orientation in relation to input product space: the input variables u and v . That is the reason why the projection of the cluster memberships on the system input give blurred membership functions. Zhao *et al.* and Babuška and Verbruggen tried to improve these blurred projections by filtering and fitting.

We propose a new method to extract membership functions used in the premise of the fuzzy model. By the definition of a cluster it is stated that the sum of the memberships u_{ij} of all clusters i for the j th feature is equal to 1:

$$\sum_{i=1}^K u_{ij} = 1, \forall j \quad (25)$$

If two clusters k_n and k_m are adjacent, then there will be features x_j with ($u_{nj} \cong 0.5$ AND $u_{mj} \cong 0.5$). All features which match this condition, can be fitted in a $n - 1$ dimensional hyperplane ($n - 1$ is the dimension of the input space). The normal of the hyperplane is the direction in which the fuzzy K -partition matrix has to be transformed. Projecting the transformed fuzzy K -partition matrix will give a clear projection. The membership function belonging to rules n and m can now be calculated, by fitting the projection into a preferred membership function (e.g. trapezoidal, exponential or sigmoidal).

The input product space of the example of Figure 3 is two-dimensional: x_1 and x_2 . This means that all features

x_j in the transition area of two clusters n and m ($u_{nj} \cong 0.5$ and $u_{mj} \cong 0.5$) must be fitted into a two-dimensional hyperplane: a line. The rotation vectors T are the normals of the fitted lines. For each of the adjacent clusters, the input product space must be transformed with the accompanying rotation vector. The rotation of a feature $x = [x_1 \ x_2]^T$ can be described as follows:

$$\tilde{x} = x^T T \quad (26)$$

in which \tilde{x} is the new projection variable. Projection of the transformed input space onto the new projection variable will result in clear projections. For the example of Figure 3 the rotation vectors are not equal, thus two rotations are needed.

Each projection can now be fitted into membership functions, one membership function for each cluster. The rulebase of this fuzzy system could look like:

$$\begin{aligned} \text{if } \tilde{x} = A_1 \text{ then } y^1 &= p_0^1 + p_1^1 x_1 + p_2^1 x_2 \\ \text{if } \tilde{x} = A_2 \text{ then } y^2 &= p_0^2 + p_1^2 x_1 + p_2^2 x_2 \\ \text{if } \tilde{x} = A_3 \text{ then } y^3 &= p_0^3 + p_1^3 x_1 + p_2^3 x_2 \end{aligned} \quad (27)$$

in which A^i are fuzzy sets corresponding to the validity regions of each cluster.

Rotation is needed for each two adjacent clusters. This means for the partition in Figure 4 (left) that the input space must be transformed twice (two boundaries). Because cluster k_2 has two adjacent clusters, the premise of that rule will contain two fuzzy sets, one for each boundary. For Figure 4 (right), this means that the input space must be transformed three times. Rules 1 and 4 will have one fuzzy set in the premise. Rules 2 and 3, however, will have three fuzzy sets in the premise.

At this stage of identification of the fuzzy model, the number of rules are known, just like the premise of the model. The consequence parameters of each rule can now be retrieved from the fuzzy K -partition matrix. A new method for calculation of these parameters will be presented in the next section.

Direct calculation method of consequence parameters

There are different ways to extract the consequence parameters from the fuzzy K -partition matrix. They can be extracted from the membership matrix U — which

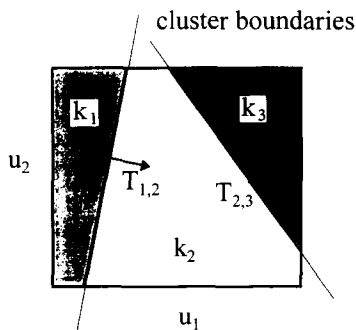


Figure 3 A realistic example of three adjacent clusters k_1 , k_2 and k_3 with their validity regions: gray areas. The normals of the cluster boundaries $T_{1,2}$ and $T_{2,3}$ are the rotation directions

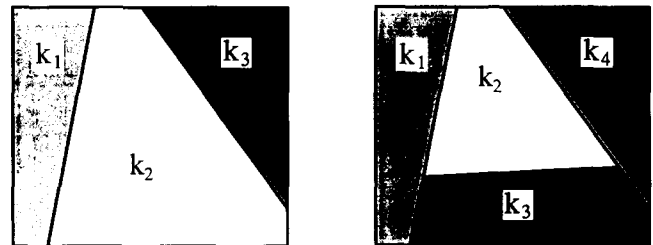


Figure 4 The fuzzy K -partitions of a two-dimensional input space, with $K = 3$ (left and 4 (right).

holds the membership of each feature x_j to each cluster k — and the data set x_j . Zhao *et al.* and Babuška and Verbruggen used a weighted least squares algorithm to fit the data, weighted with the membership, into a hyperplane. It is questionable if it is necessary to use a fitting algorithm twice. The fuzzy GK clustering can also be interpreted as a fitting algorithm. This is a good reason to propose a new method for calculation of the consequence parameters, based on the clustering result only. The following section will demonstrate that the new method is not only simpler but also faster if large data sets are used.

The consequence of the i th rule of the rule base

$$y^i = p_1^i + p_2^i u_1^i + \dots + p_n^i u_{n-1}^i \quad (28)$$

can be considered as a n -dimensional hyperplane. Let, for each i th cluster, F_i be the covariance matrix, and v_i the cluster center. Let $\{\phi_{i,1}, \dots, \phi_{i,n}\}$ be the eigenvectors of the covariance matrix of the i th cluster, arranged in descending order. The principal axis of the hyperplane will be given approximately by the $n - 1$ largest eigenvectors. To calculate the offset or bias term p_1 , a point on the hyperplane is needed. There is only one point that lies exactly on the i th hyperplane: the i th cluster center v_i . Basically we have n variables and n unknown parameters, thus n points on the hyperplane are needed. To retrieve those points the eigenvectors can be added to the cluster center:

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} u_{1,1} & \dots & u_{n-1,1} & y_1 \\ u_{1,2} & \dots & u_{n-1,2} & y_2 \\ \vdots & \ddots & \vdots & \vdots \\ u_{1,n} & \dots & u_{n-1,n} & y_n \end{bmatrix} \\ &= \begin{bmatrix} v_1 + \phi_{1,1} & v_2 + \phi_{1,2} & \dots & v_n + \phi_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ v_1 + \phi_{n-1,1} & v_2 + \phi_{n-1,2} & \dots & v_n + \phi_{n-1,n} \\ v_1 & v_2 & \dots & v_n \end{bmatrix} \end{aligned} \quad (29)$$

Equation (28) can be rewritten for the i th rule as follows:

$$y^i = \mathbf{u}^i \mathbf{p}^i \quad (30)$$

in which $\mathbf{p}^i = [p_1, \dots, p_n]$ are the consequence parameters, y^i the output of the i th rule and $\mathbf{u}^i = [1, u_1, \dots, u_{n-1}]^T$ the input variables. Writing Equation (30) for n equations gives:

$$\mathbf{y}^i = \mathbf{U}^i \mathbf{p}^i \quad (31)$$

in which

$$\mathbf{U}^i = \begin{bmatrix} 1 & u_{1,1} & \dots & u_{n-1,1} \\ 1 & u_{1,2} & \dots & u_{n-1,2} \\ 1 & \vdots & \ddots & \vdots \\ 1 & u_{1,n} & \dots & u_{n-1,n} \end{bmatrix} \text{ and } \mathbf{y}^i = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

This equation can be solved for \mathbf{p}^i by inverting the input matrix \mathbf{U}^i :

$$\mathbf{p}^i = (\mathbf{U}^i)^{-1} \mathbf{y}^i \quad (32)$$

By using the datapoints x_j^i of Equation (29) as u_j^i in Equation (32), the consequence parameters \mathbf{p}^i can be solved:

$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{bmatrix} = \begin{bmatrix} 1 & \phi_{1,1} + v_1 & \dots & \phi_{n-1,1} + v_{n-1} \\ 1 & \vdots & \ddots & \vdots \\ 1 & \phi_{n-1,1} + v_1 & \dots & \phi_{n-1,n-1} + v_{n-1} \\ 1 & v_1 & \dots & v_{n-1} \end{bmatrix}^{-1} \begin{bmatrix} \phi_{1,n} + v_n \\ \vdots \\ \phi_{n-1,n} + v_n \\ v_n \end{bmatrix} \quad (33)$$

Optimization of model parameters

Many papers about fuzzy modeling use optimization of model parameters. Often a least square method is used. Optimization of consequence parameters may give a perfect global fit but can also lead to a bad local representation of the system. In this case a minor change in the premise parameters may give a major change in the consequence parameters.

Optimization of premise parameters can lead to crisp boundaries of rules instead of fuzzy boundaries. The fuzzy clustering algorithm is already an optimization algorithm, for these reasons we do not use optimization of the model parameters.

Summary of the identification algorithm

1. choose the process input variables and output variable
2. choose the model orders n, m and delay d
3. choose the maximum number of fuzzy rules, thus the maximum number of clusters k_{\max}
4. perform GK hyperplane clustering [Equations (15)–(19)] and compatible cluster merging [Equations (20)–(24)]. Choose the CCM threshold γ
5. perform the cluster rotation technique
6. project the rotated input space onto the (transformed) input variables
7. calculate the consequence parameters using the direct method.

Simulations

The proposed technique has been applied to two processes: a nonlinear dynamic system described by Zhao *et al.*⁹ and a bioreactor described by Agrawal *et al.*²².

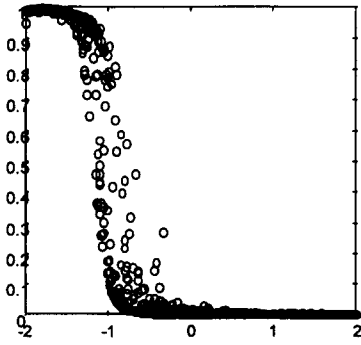
The first model is given by:

$$y(k) = y(k-1)^3 - 0.2|y(k-1)|u(k-1) + 0.08u(k-1)^2 \quad (34)$$

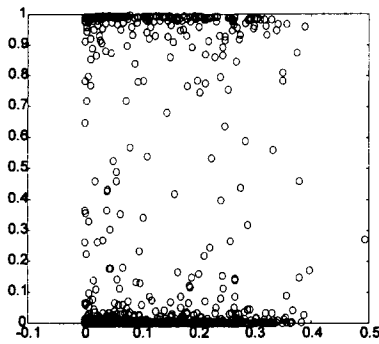
Let the input signal u for identification be a uniformly distributed noise in the interval $[-2, 2]$, the number of sample data $N = 1001$, and assume the number of fuzzy rules is not known *a priori*. First the GK clustering is carried out with eight clusters, which is much more than the three clusters Zhao *et al.* needed to identify their fuzzy model of this system. The cluster merging technique is performed with a threshold value of 0.8.

The clustering software computes the fuzzy K -partition matrix U (a matrix with membership values of each feature to each cluster), and the consequence parameters p_i for each rule i . By projecting the fuzzy K -partition matrix on the input variables $u(k-1)$ and $y(k-1)$, an initial fuzzy model with four rules is obtained:

R^1 : if $u(k-1)$ is

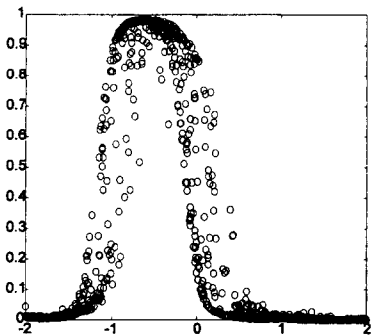


and $y(k-1)$ is

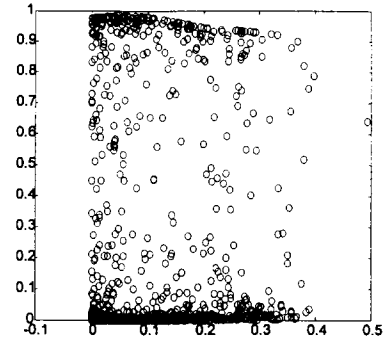


then $y^1(k) = -0.225 - 0.269 * u(k-1) + 0.403 * y(k-1)$

R^2 : if $u(k-1)$ is

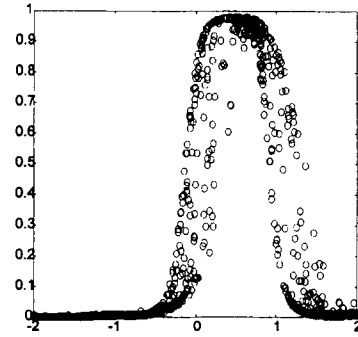


and $y(k-1)$ is

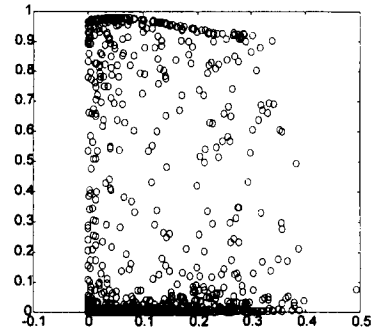


then $y^2(k) = -0.031 - 0.103 * u(k-1) + 0.188 * y(k-1)$

R^3 : if $u(k-1)$ is

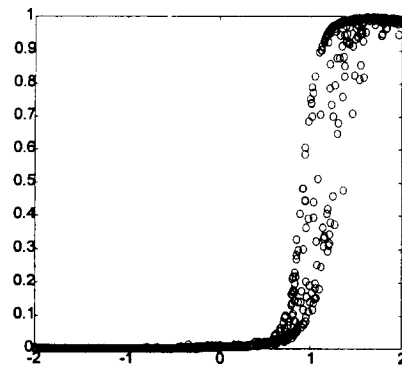


and $y(k-1)$ is

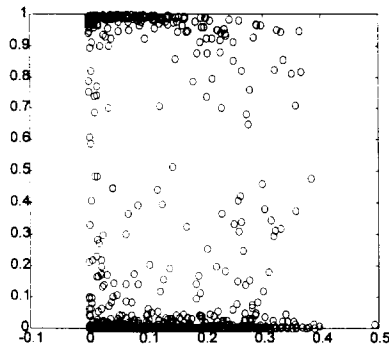


then $y^3(k) = -0.007 + 0.060 * u(k-1) - 0.031 * y(k-1)$

R^4 : if $u(k-1)$ is



and $y(k-1)$ is



then $y^4(k) = -0.154 + 0.224 \cdot u(k-1) - 0.233 \cdot y(k-1)$

It can be seen that for the rules, the fuzzy sets of $y(k-1)$ are 'flat'. This implies that $y(k-1)$ has no meaningful influence on the rules and can be removed. It is clearly visible that rotation of the fuzzy partition matrix is necessary, because the memberships $u(k-1)$ are blurred. These blurred projections are caused by the orientation of the clusters with respect to the input variables. These orientations are visualized in Figures 5 and 6. These figures show also the four validity regions of the clusters, in which the validity is represented by the membership values in the interval $[0, 1]$.

Removing the non-necessary input variables with 'flat fuzzy sets' from the premise parts cannot be viewed as a strict approximation. In a sense, it is a heuristic method since the distribution of the input-output samples are not considered. These flat fuzzy sets can be regarded as the fuzzy label ANY (instead of HIGH or VERY LOW), which is not a restriction.

A linear regression method is used to fit all features with membership values between $\mu = 0.4$ and $\mu = 0.6$ into borderlines between the adjacent clusters. The values 0.4 and 0.6 are used because one needs enough features to determine the rotation directions. The results of the linear regression algorithm are three directions of the borderlines of the adjacent clusters:

$$\begin{aligned} T_{1,2} &= \begin{pmatrix} 1 \\ 0.79 \end{pmatrix}, \quad r = 0.992 \\ T_{2,3} &= \begin{pmatrix} 1 \\ 0.71 \end{pmatrix}, \quad r = 0.991 \\ T_{3,4} &= \begin{pmatrix} 1 \\ 0.86 \end{pmatrix}, \quad r = 0.995 \end{aligned} \quad (35)$$

where T_{ij} is a vector which gives the direction of the borderline between adjacent i th and the j th clusters and r is the correlation coefficient for the linear regression. Hence the rotation directions are the normals of the vectors T_{ij} .

Thus the fuzzy partition matrix must be transformed, by creating a new set of input variables \tilde{u}_i . The rotation is done by multiplying the product input space with the normals of vectors T_{ij} :

$$\begin{aligned} \tilde{u}_1 &= \begin{pmatrix} u(k-1) \\ y(k-1) \end{pmatrix} \begin{pmatrix} -0.79 & 1 \end{pmatrix}, \\ \tilde{u}_2 &= \begin{pmatrix} u(k-1) \\ y(k-1) \end{pmatrix} \begin{pmatrix} -0.71 & 1 \end{pmatrix}, \\ \tilde{u}_3 &= \begin{pmatrix} u(k-1) \\ y(k-1) \end{pmatrix} \begin{pmatrix} -0.86 & 1 \end{pmatrix} \end{aligned} \quad (36)$$

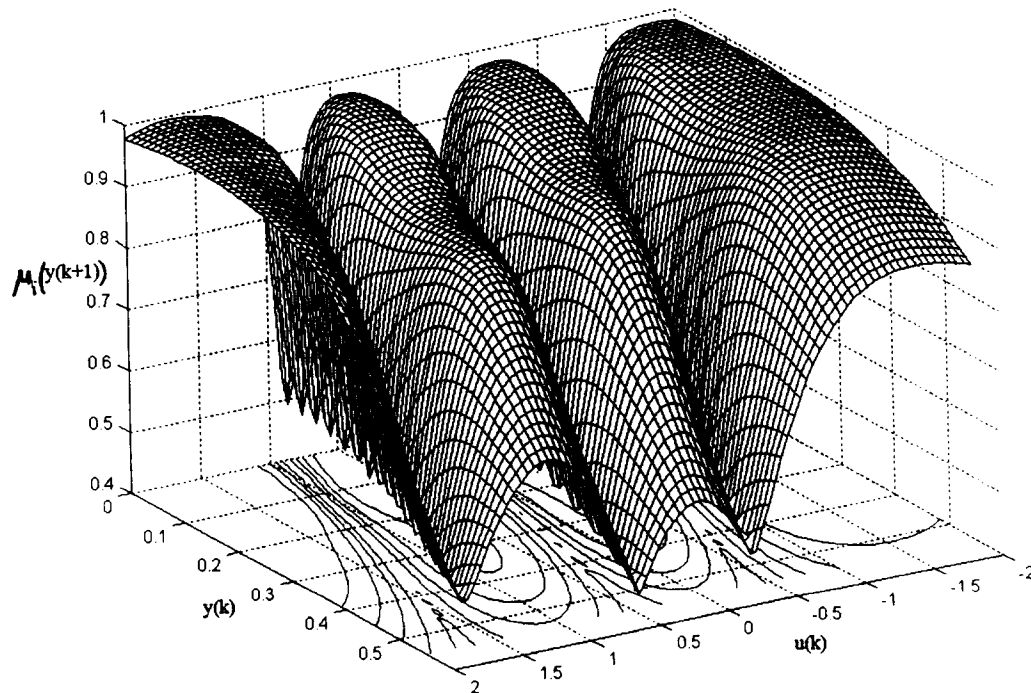


Figure 5 Fuzzy K -partitions with orientation. The orientations are clearly visible from the contour plot on the bottom plane

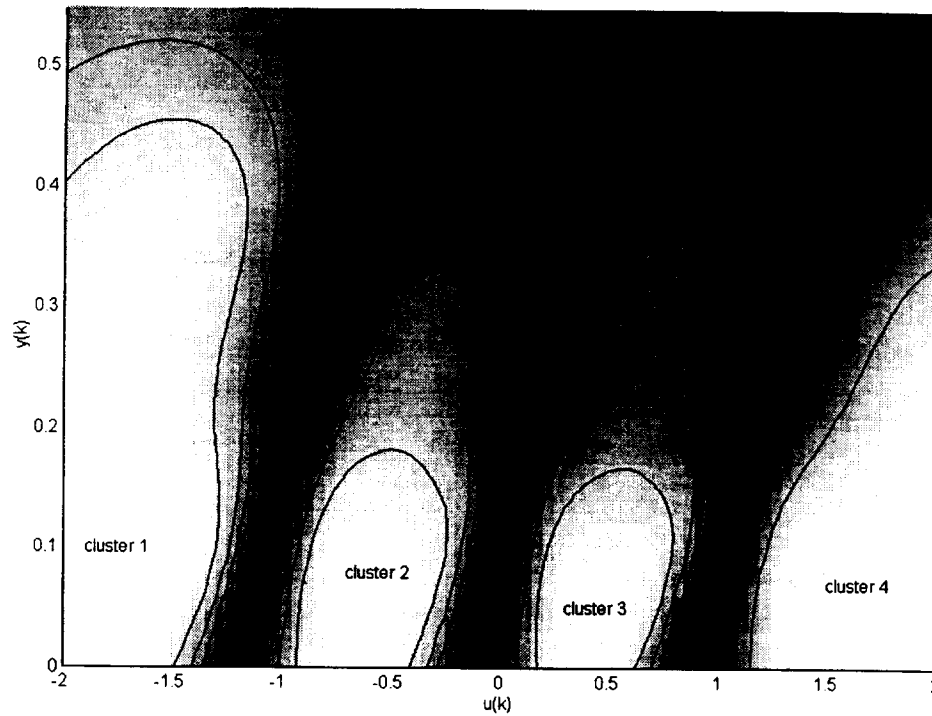
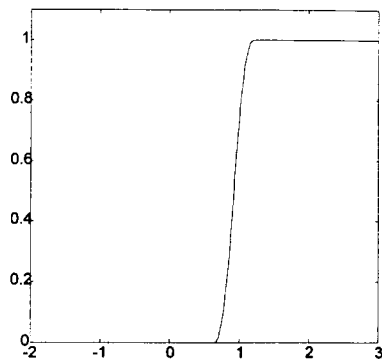


Figure 6 Contour plot of the clusters membership values. The cluster have an orientation with respect to the input variables $u(k)$ and $y(k)$

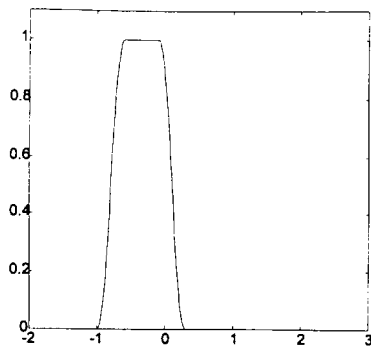
Then by removing the membership functions of $y(k)$ and replacing $u(k-1)$ with \tilde{u}_1 , and fitting the membership functions for \tilde{u}_1 into sigmoidal shapes, one obtains the following fuzzy model:

R^1 : if \tilde{u}_1 is

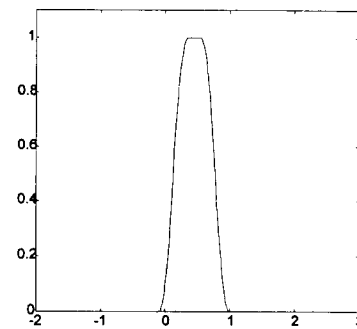


then $y^1(k) = -0.225 - 0.269 * u(k-1) + 0.403 * y(k-1)$

R^2 : if \tilde{u}_1 is

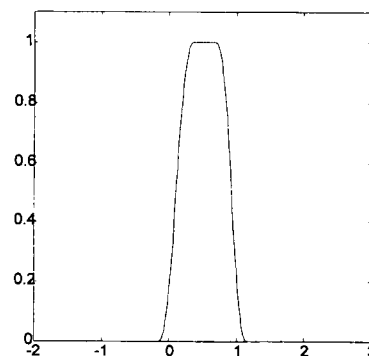


and \tilde{u}_2 is

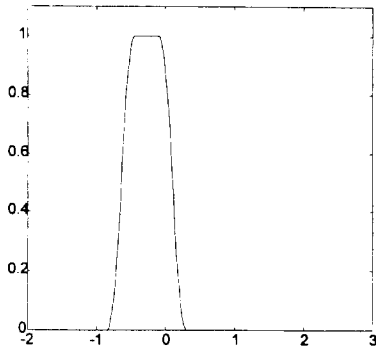


then $y^2(k) = -0.031 - 0.103 * u(k-1) + 0.188 * y(k-1)$

R^3 : if \tilde{u}_2 is

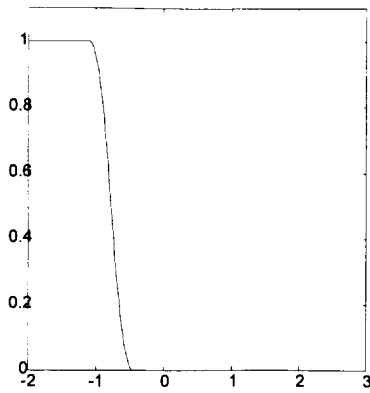


and u_3 is



then $y^3(k) = -0.007 + 0.060 * u(k-1) - 0.031 * y(k-1)$

R^4 : if \tilde{u}_3 is



then $y^4(k) = -0.154 + 0.224 * u(k-1) - 0.233 * y(k-1)$

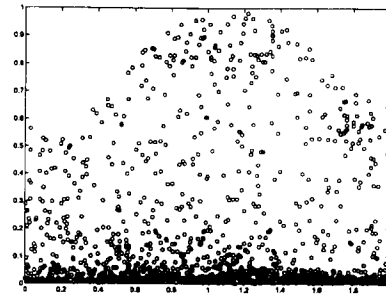
This fuzzy model is validated by using an input signal, which is composed of some steps, noise and a first order time lag. Figure 7 shows the responses of the system and the final fuzzy model.

The performance of the new identification method compared to the identification method proposed by Zhao *et al.*⁹ is shown in Table 1.

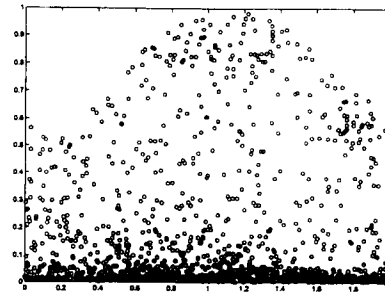
The second system on which the fuzzy modeling approach was applied was the bioreactor described by Agrawal *et al.*²². This process is extremely nonlinear. A detailed description of the reactor is given in Appendix II. An identification data set was created, consisting of 1500 features, uniformly and randomly distributed. The domain of the input variables were: $w(k) = [0, 2]$, $x_1(k) = [0, 1]$, $x_2(k) = [0, 1]$.

After fuzzy clustering, an initial model is obtained, consisting of seven rules. The cluster merging technique was performed with a threshold value of 0.35. Compared to the Zhao *et al.* case, this is a low threshold. It was chosen in order to reduce the number of clusters. The initial model is described by:

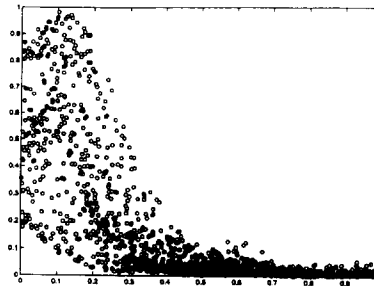
R^1 : IF $w(k)$ is



AND $x_2(k)$ is

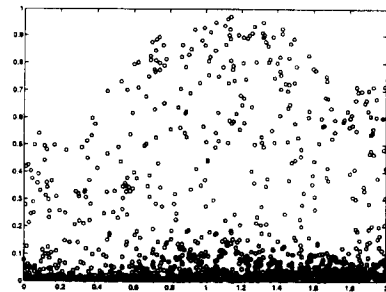


AND $x_1(k)$ is



THEN $\Delta x_1^1(k) = 0.999 - 0.101 * w(k) + 0.0197 * x_2(k) + 0.228 * x_1(k)$

R^2 : IF $w(k)$ is



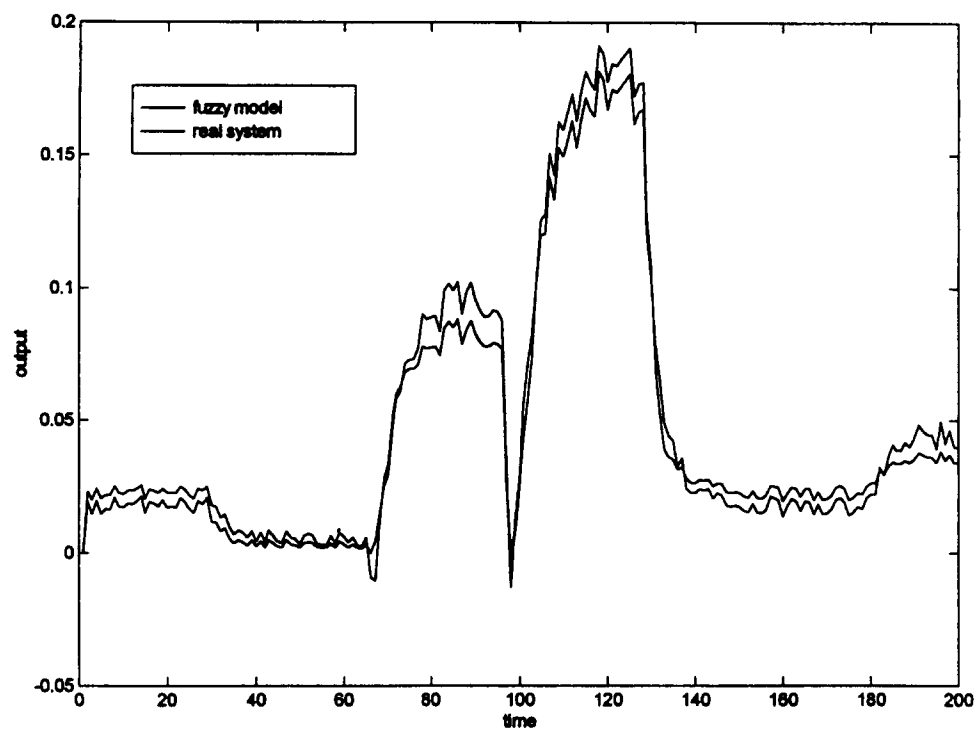


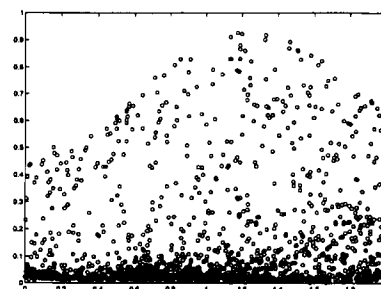
Figure 7 Output signals of the fuzzy model (y_k) and the real system

Table 1 Performance of the fuzzy models using the new identification method in comparison with the identification method used by Zhao *et al.*

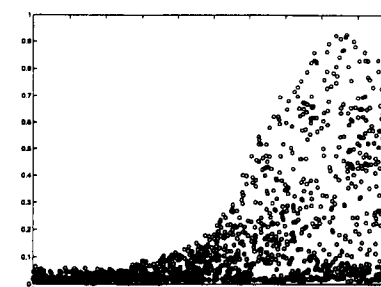
Method	# rules	Dataset	Model used	Model error ε
Zhao	3	identification	initial	2.2×10^{-4}
		identification	final (optimized)	1.6×10^{-4}
		identification	final (optimized)	1×10^{-4}
New	4	identification	with rotation	0.75×10^{-4}
		validation	with rotation	0.50×10^{-4}

$$\text{THEN } \Delta x_1^2(k) = 0.796 - 0.864 * w(k) + 0.0698 * x_2(k) + 0.140 * x_1(k)$$

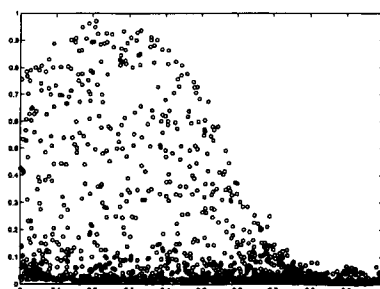
R^3 : IF $w(k)$ is



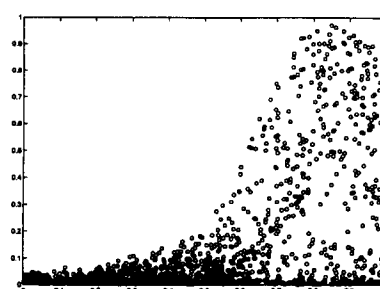
AND $x_2(k)$ is



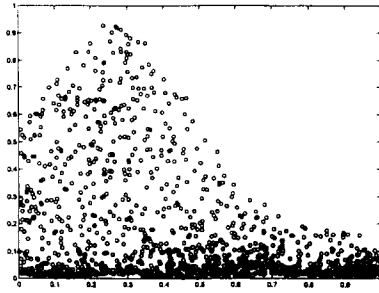
AND $x_2(k)$ is



AND $x_1(k)$ is

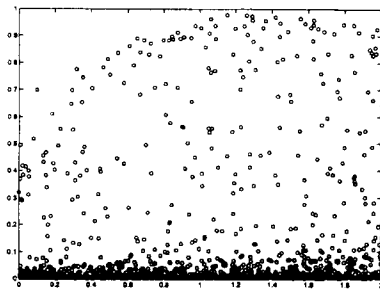


AND $x_1(k)$ is

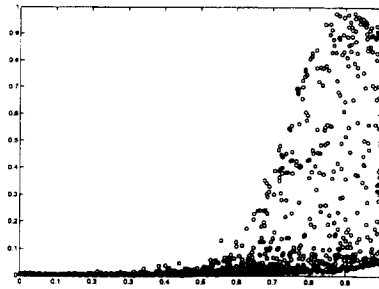


THEN $\Delta x_1^3(k) = 1.07 - 0.0240 * w(k) - 0.0936 * x_2(k) - 0.416 * x_1(k)$

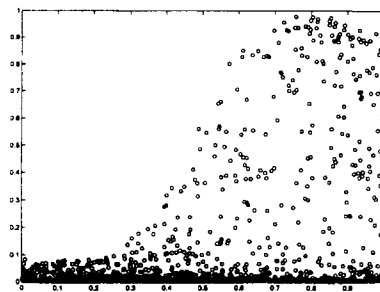
R^4 : IF $w(k)$ is



AND $x_2(k)$ is

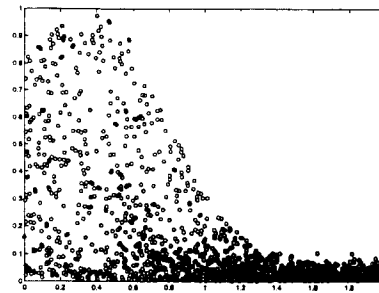


AND $x_1(k)$ is

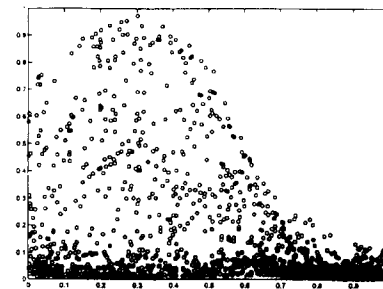


THEN $\Delta x_1^4(k) = 4.66 - 0.800 * w(k) - 3.97 * x_2(k) - 0.740 * x_1(k)$

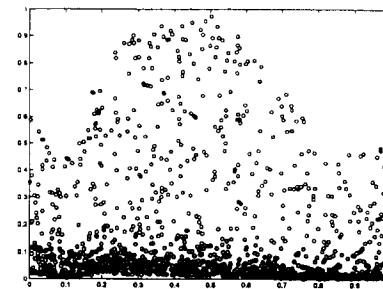
R^5 : IF $w(k)$ is



AND $x_2(k)$ is

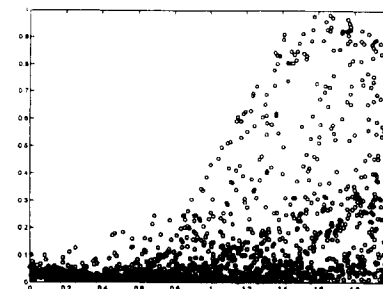


AND $x_1(k)$ is

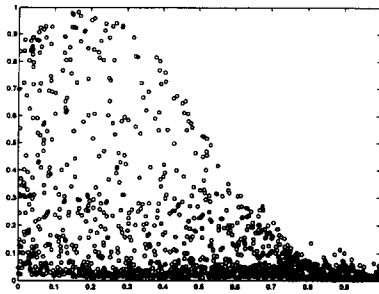


THEN $\Delta x_1^5(k) = 0.0454 - 0.433 * w(k) + 0.271 * x_2(k) + 0.984 * x_1(k)$

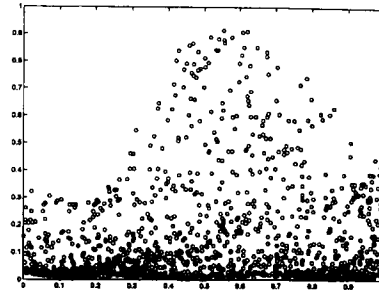
R^6 : IF $w(k)$ is



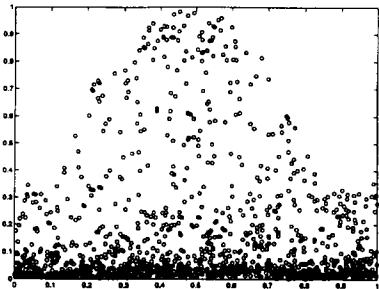
AND $x_2(k)$ is



AND $x_1(k)$ is

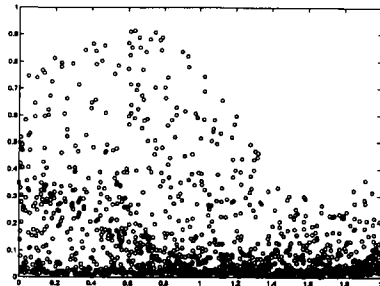


AND $x_1(k)$ is

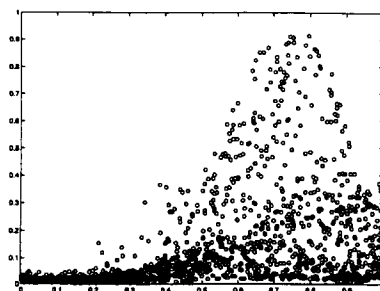


$$\text{THEN } \Delta x_1^6(k) = 0.756 - 0.486 * w(k) + 0.389 * x_2(k) - 0.526 * x_1(k)$$

R^7 : IF $w(k)$ is



AND $x_2(k)$ is



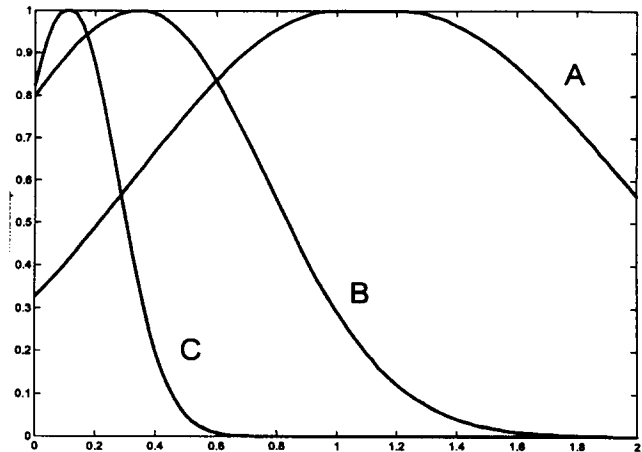
$$\text{THEN } \Delta x_1^7(k) = 1.19 - 0.598 * w(k) - 1.17 * x_2(k) + 0.603 * x_1(k)$$

The premise parameters are fitted into sigmoidal functions, as a result of which the following fuzzy sets are found for the premise parameters

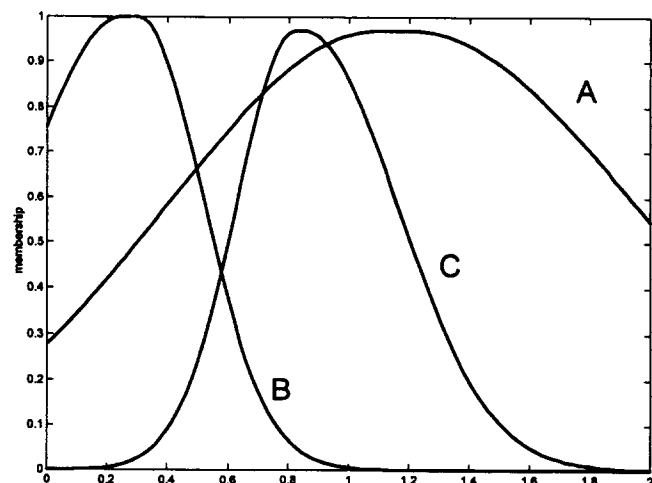
IF $w(k)$ is A^i AND $x_2(k)$ is B^i AND $x_1(k)$ is C^i THEN ...

in which

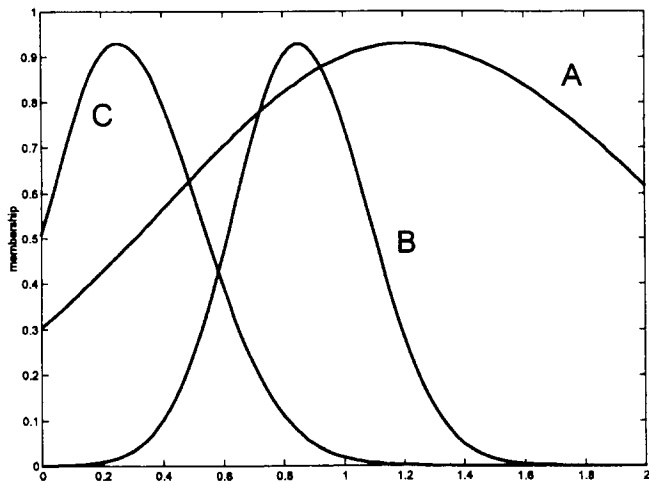
rule 1:



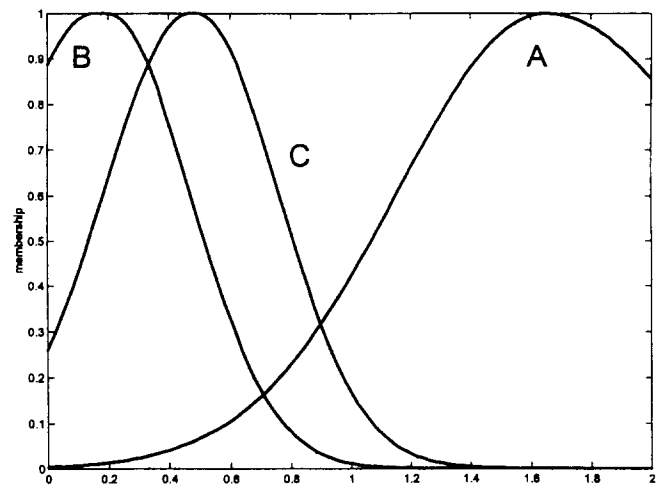
rule 2:



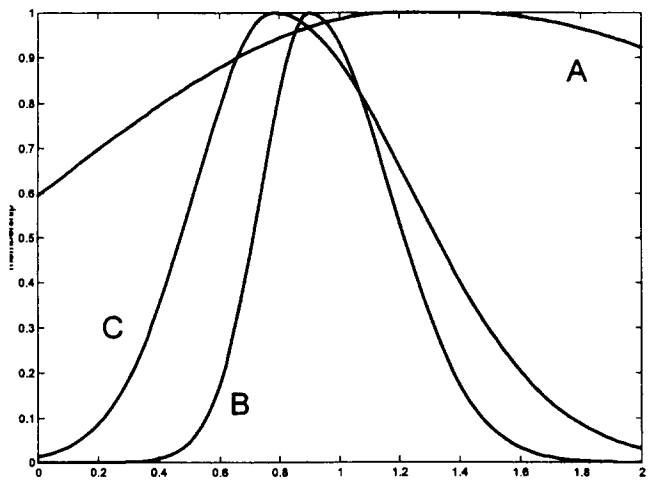
rule 3:



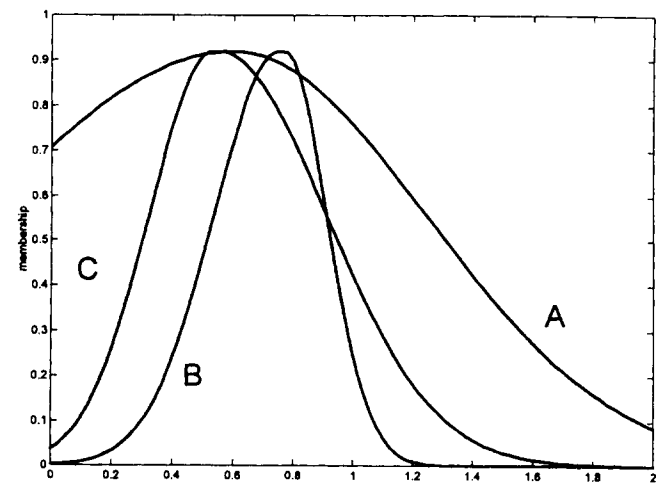
rule 6:



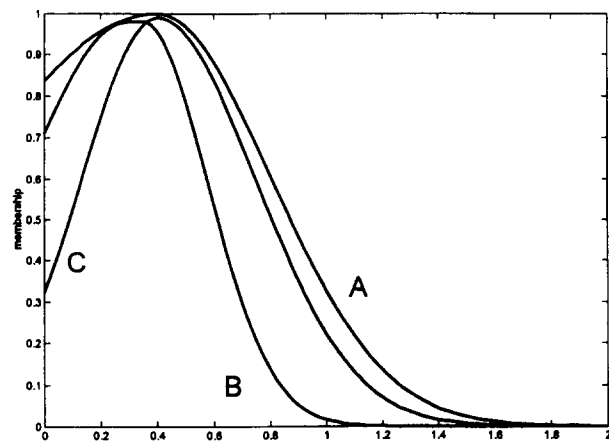
rule 4:



rule 7:



rule 5:



The output of the nonlinear bioreactor and the fuzzy model for a variation in the reactor input are shown in Figure 8. In this case it was difficult to find the transformation vectors, due to the extreme nonlinearity of the model, and the limited number of rules which generated to describe this model. In addition, more research will be needed to develop a systematic approach for finding transformation vectors in more-dimensional systems.

Comparison to related work

In the foregoing sections reference was already made to the work of Takagi and Sugeno⁷, Babuška and Kaymak⁸ and Zhao *et al.*⁹. Also the differences between these authors and the present investigations were highlighted. Johansen and Foss²³ recently published a semi-empirical modeling approach to nonlinear dynamic systems through identification of operating regimes and local models. In this approach, the operating space is decomposed into operating regimes and variables which

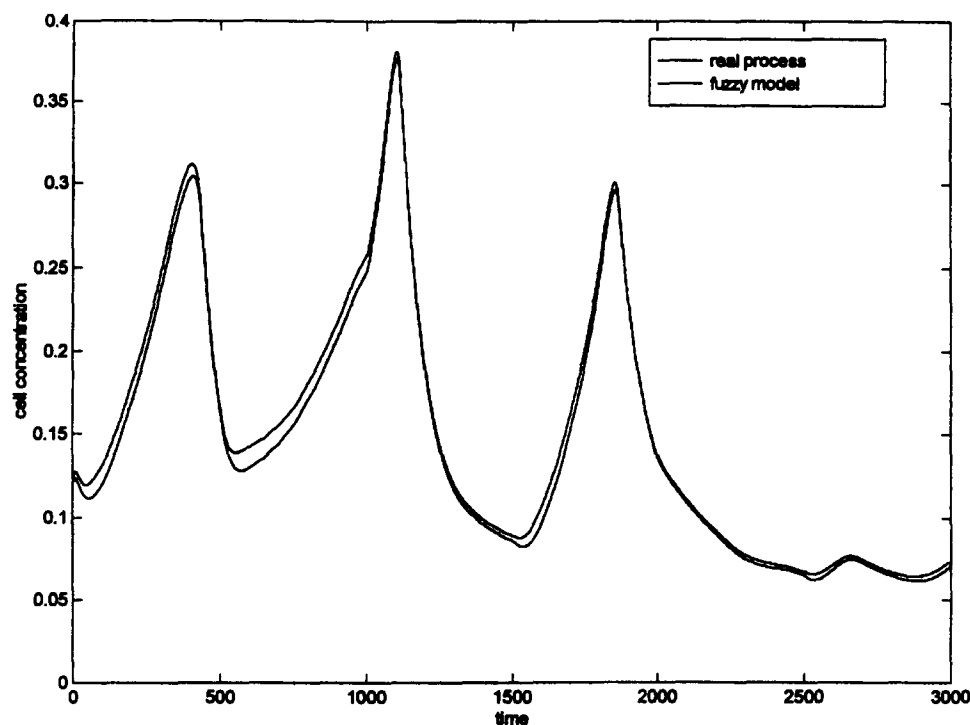


Figure 8 The output signals of the fuzzy model and the original nonlinear model

characterize each operating regime are chosen. In addition, local model structures are chosen which may be based on e.g. mass and energy balances, hence *a priori* knowledge can be embedded in the local models. Then the local model parameters are identified for all regimes and interpolation functions are chosen for providing a smooth interpolation between the local modes. A typical choice for the interpolation function are kernel functions like the unnormalized Gaussian exponential function.

In our work no operating regimes are pre-defined, and no *a priori* process knowledge is required.

Conclusions

An identification method for fuzzy models was proposed and tested on two processes. In the method two new features were proposed: membership rotation and direct calculation of the consequence parameters. Membership rotation has the advantage that membership functions can be extracted which are less blurred, thus more accurately defined. The direct calculation of the consequence parameters has the advantage that the additional minimization step of fitting the data, weighted with the membership, into a hyperplane, can be omitted. As a result, the newly proposed method leads to fuzzy models which describes the real system behavior more accurately. The major advantage of the fuzzy models over a neural network model or the original nonlinear model is its relative simplicity. In addition the fuzzy model has fewer parameters than a neural network and is structurally more attractive.

References

- 1 Tong, R. M. in 'Advances in Fuzzy Sets Theory and its Applications' (Eds M. M. Gupta *et al.*), North Holland, New York 1979, 559–576
- 2 Pedrycz, W. *Fuzzy Sets and Systems* 1984, **13**, 153
- 3 Yi, S. Y. and Chung, M. J. *Fuzzy Sets and Systems* 1993, **59**, 25
- 4 Babuška, R. and Verbruggen, H. B. in 'Proceedings IFAC Congress AIRTC '94', Valencia, Spain, 3–5 October 1994
- 5 Zadeh, L. A. *Inf. Sci.* 1975, **8**, 199
- 6 Sugeno, M. and Yasukawa, T. *IEEE Trans. Fuzzy Systems* 1993, **1**(1), 7
- 7 Takagi, T. and Sugeno, M. *IEEE Trans. Systems Man Cybernet.* 1985, **15**, 166
- 8 Babuška, R. and Kaymak, U. 'Proc. European Congress on Fuzzy and Int. Tech.', Aachen, Germany 1995, 565
- 9 Zhao, J., Wertz, V. and Gorez, R. 'Proc. 9th IEEE Int. Symp. on Int. Control', Columbus, Ohio, USA, 1994, 172
- 10 Babuška, R. and Verbruggen, H. B. 'Proc. IFAC Congress SICICA '94', Budapest, 1994
- 11 Zadeh, L. A. *Inform. Control* 1965, **8**, 338
- 12 Mamdani, E. H. *Proc. IEE* 1974, **121**(12), 1585
- 13 Yoshinari, Y., Pedrycz, W. and Hirota, K. *Fuzzy Sets Systems* 1993, **54**, 157
- 14 Yager, R. R. and Filev, D. P. *IEEE Trans. Systems, Man and Cybernet.* 1993, **23**, 1198
- 15 Zhao, J., Gorez, R. and Wertz, V. *Automatica*, 1994, submitted
- 16 Berenji, H. R. and Khedkar, P. S. 'Proc. of the 2nd IEEE Int. Conf. on Fuzzy Systems', 1993, 1402
- 17 Gustafson, D. E. and Kessel, W. C. 'IEEE Conf. on Decision and Control', 1979, 761
- 18 Sin, S. and deFigueiredo, R. J. P. 'Proc. of the 2nd IEEE Int. Conf. on Fuzzy Systems', 1993, 190
- 19 Windham, M. P. *Fuzzy Sets and Systems* 1983, **10**, 271
- 20 Kaymak, U. and Babuška, R. 'FUZZ-IEEE/IFES '95' Yokohama, Japan, March 1995
- 21 Yang, M. S. *Math. Comput. Modelling* 1993, **18**(11), 1
- 22 Agrawal, P., Lee, C., Lim, H. C. and Ramkrishna, D. *Chem. Engng Sci.* 1982, **37**(3), 453
- 23 Johansen, T. A. and Foss, B. *Modeling, Identification and Control* 1995, **16**(4), 213

Appendix I: Function approximation concept

Consider a first order nonlinear dynamic system described by:

$$y(k+1) = y(k) + u(k) \cdot e^{-3|y(k)|} \quad (\text{I.1})$$

For a random sequence $u(k)$, the output $y(k)$ will be measured. After estimation of m , n and d , which are in this case $m = n = 1$ and $d = 0$, an initial dataset can be created:

$$D = \begin{bmatrix} u(1) & y(1) \\ u(2) & y(2) \\ \vdots & \vdots \\ u(N-1) & y(N-1) \end{bmatrix} \quad Y = \begin{bmatrix} y(2) \\ y(3) \\ \vdots \\ y(N) \end{bmatrix} \quad (\text{I.2})$$

The dynamic system can be described as:

$$Y = f(D) \quad (\text{I.3})$$

The function f describes the relation between the regression matrix D (the measurements) and the output of the systems Y . The geometric interpretation of Equation (I.3) is that system describes a subspace in $\mathbb{R}^{(m+n+1)}$ in which f describes a hypersurface. For this system this means that the function f is a surface in \mathbb{R}^3 (See Figure I.1)

Identification of the nonlinear dynamic system means approximation of this surface.

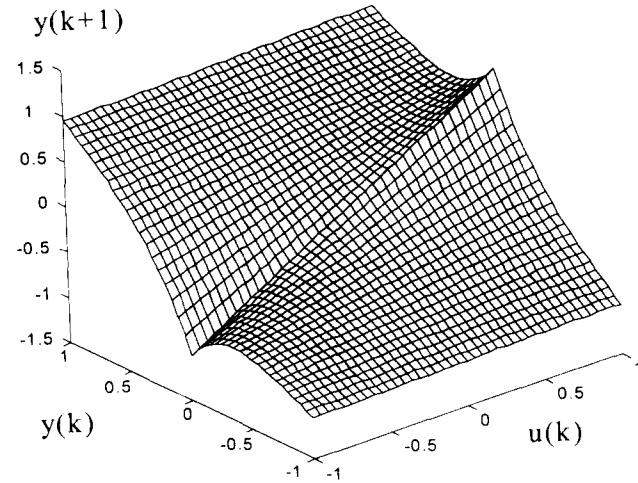


Figure I.1 The surface in \mathbb{R}^3 of the system [Equation (I.1)]

$$\begin{aligned} x_1(k+1) &= x_1(k) \\ &+ \Delta(-x_1(k) \cdot w(k) + x_1(k) \cdot (1 - x_2(k)) \cdot e^{x_2(k)/\gamma}) \end{aligned} \quad (\text{II.3})$$

$$\begin{aligned} x_2(k+1) &= x_2(k) \\ &+ \Delta(-x_2(k) \cdot w(k) \\ &+ x_1(k) \cdot (1 - x_2(k)) \cdot e^{x_2(k)/\gamma} \frac{1 + \beta}{1 + \beta - x_2}) \end{aligned} \quad (\text{II.4})$$

Appendix II: Bioreactor model

In this appendix the derivation of dimensionless equations from the mass balances is illustrated, used to describe a bioreactor²². The cell concentration and the substrate concentration can be described as:

$$\frac{dX}{dt} = -\frac{FX}{V} + \mu(S)X \quad (\text{II.1})$$

$$\frac{dS}{dt} = \frac{F(S_f - S)}{V} - \sigma(S)X \quad (\text{II.2})$$

Cell and substrate concentrations and time can be made dimensionless²² and discretized, resulting in the following set of equations:

Bioreactor model notation

w	flow rate (hour ⁻¹) = [0, 2]
F	flow rate (m ³ /hour)
S	substrate concentration (kg/m ³)
S_f	feed substrate conc. (constant) (kg/m ³)
t	independent variable time (hour)
V	reactor volume (constant) (m ³)
X	cell concentration (kg/m ³)
β	growth rate parameter [$\cdot 0$] = 0.02
Δ	sampling time (hour) = 0.01
γ	substrate inhibition parameter [$\cdot 0$] = 0.48
μ	specific growth rate (hour ⁻¹)
x_1	normalized cell concentration [$\cdot 0$] = [0, 1]
x_2	substrate conversion ($S_f - S$)/ S_f , [$\cdot 0$] = [0, 1]
σ	substrate consumption rate (hour ⁻¹)